

# Package ‘gnlm’

May 6, 2025

**Version** 1.1.2

**Title** Generalized Nonlinear Regression Models

**Depends** R (>= 1.4), rmutl

**Description** A variety of functions to fit linear and nonlinear regression with a large selection of distributions.

**License** GPL (>= 2)

**URL** <https://www.commanster.eu/rcode.html>

**BugReports** <https://github.com/swihart/gnlm/issues>

**Encoding** UTF-8

**RoxygenNote** 7.2.1

**NeedsCompilation** yes

**Author** Bruce Swihart [cre, aut],  
Jim Lindsey [aut] (Jim created this package, Bruce is maintaining the CRAN version)

**Maintainer** Bruce Swihart <bruce.swihart@gmail.com>

**Repository** CRAN

**Date/Publication** 2025-05-06 16:10:05 UTC

## Contents

bnlr . . . . .	2
fit.dist . . . . .	4
fmr . . . . .	6
gnlr . . . . .	10
gnlr3 . . . . .	14
nlr . . . . .	18
nordr . . . . .	20
ordglm . . . . .	23
rs2 . . . . .	24
rs3 . . . . .	25

<b>Index</b>	<b>27</b>
--------------	-----------

**Description**

bnlr fits user-specified nonlinear regression equations to binomial data with various link functions (logit, probit, comp log log, log log, Cauchy, Student t, stable, or mixture). The mixture link is a logistic link with extra probability mass for  $y=0$  and  $y=n$ .

**Usage**

```
bnlr(
  y = NULL,
  link = "logit",
  mu = NULL,
  linear = NULL,
  pmu = NULL,
  pshape = NULL,
  wt = 1,
  envir = parent.frame(),
  print.level = 0,
  typsize = abs(p),
  ndigit = 10,
  gradtol = 1e-05,
  stepmax = 10 * sqrt(p %% p),
  steptol = 1e-05,
  iterlim = 100,
  fscale = 1
)
```

**Arguments**

- |      |  |
|------|--|
| y    | A two column matrix of binomial data or censored data or an object of class, response (created by <a href="#">restovec</a> ) or repeated (created by <a href="#">rmna</a> or <a href="#">lvna</a> ). If the repeated data object contains more than one response variable, give that object in <code>envir</code> and give the name of the response variable to be used here.                                  |
| link | A character string containing the name of the link function. The Student t, stable, and mixture links contain an unknown parameter to be estimated, respectively the logarithm of the degrees of freedom, the tail parameter transformed by $\log(\text{tail}/(2-\text{tail}))$ , and logit of the mixture probability, so that they lie on the whole real line.   |
| mu   | A user-specified function of pmu, and possibly linear, giving the regression equation for the location. This may contain a linear part as the second argument to the function. It may also be a formula beginning with <code>~</code> , specifying either a linear regression function for the location parameter in the Wilkinson and Rogers notation or a general function with named unknown parameters. If |

	it contains unknown parameters, the keyword <code>linear</code> may be used to specify a linear part. If nothing is supplied, the location is taken to be constant unless the linear argument is given.
<code>linear</code>	A formula beginning with <code>~</code> in W&R notation, specifying the linear part of the regression function for the location parameter or list of two such expressions for the location and/or shape parameters.
<code>pmu</code>	Vector of initial estimates for the location parameters. If <code>mu</code> is a formula with unknown parameters, their estimates must be supplied either in their order of appearance in the expression or in a named list.
<code>pshape</code>	If the link is Student t, an initial estimate of the degrees of freedom; if it is stable, an estimate of the tail parameter; if it is mixture, an estimate of the mixture probability.
<code>wt</code>	Weight vector.
<code>envir</code>	Environment in which model formulae are to be interpreted or a data object of class, <code>repeated</code> , <code>tccov</code> , or <code>tvcov</code> ; the name of the response variable should be given in <code>y</code> . If <code>y</code> has class <code>repeated</code> , it is used as the environment.
<code>print.level</code>	Arguments controlling <a href="#">nlm</a> .
<code>typsize</code>	Arguments controlling <a href="#">nlm</a> .
<code>ndigit</code>	Arguments controlling <a href="#">nlm</a> .
<code>gradtol</code>	Arguments controlling <a href="#">nlm</a> .
<code>stepmax</code>	Arguments controlling <a href="#">nlm</a> .
<code>steptol</code>	Arguments controlling <a href="#">nlm</a> .
<code>iterlim</code>	Arguments controlling <a href="#">nlm</a> .
<code>fscale</code>	Arguments controlling <a href="#">nlm</a> .

### Details

Nonlinear regression models can be supplied as formulae where parameters are unknowns in which case factor variables cannot be used and parameters must be scalars. (See [finterp](#).)

The printed output includes the `-log` likelihood (not the deviance), the corresponding AIC, the maximum likelihood estimates, standard errors, and correlations.

### Value

A list of class `gnlm` is returned that contains all of the relevant information calculated, including error codes.

### Author(s)

J.K. Lindsey

### See Also

[finterp](#), [glm](#), [gnlr](#), [gnlr3](#)

**Examples**

```

# assay to estimate LD50
y <- c(9,9,10,4,1,0,0)
y <- cbind(y,10-y)
dose <- log10(100/c(2.686,2.020,1.520,1.143,0.860,0.647,0.486))

summary(glm(y~dose, family=binomial))
bnlr(y, mu~dose, pmu=c(1,1))
summary(glm(y~dose, family=binomial(link=probit)))
bnlr(y, link="probit", mu~dose, pmu=c(1,1))
## Not run:
bnlr(y, link="log log", mu~dose, pmu=c(1,1))
bnlr(y, link="comp log log", mu~dose, pmu=c(1,1))
bnlr(y, link="Cauchy", mu~dose, pmu=c(60,-30))
bnlr(y, link="Student", mu~dose, pmu=c(60,-30), pshape=0.1)
bnlr(y, link="stable", mu~dose, pmu=c(20,-15), pshape=0, stepmax=1)
bnlr(y, link="mixture", mu~dose, pmu=c(60,-30), pshape=-2.5)
#
mu <- function(p) -p[1]*(log10(p[2])-dose)
bnlr(y, mu=mu, pmu=c(1,100))
bnlr(y, link="probit", mu=mu, pmu=c(1,100))

## End(Not run)

```

fit.dist

*Fit Probability Distributions to Frequency Data***Description**

fit.dist fits the distributions in Chapter 4 of Lindsey (1995, 2003 2nd edn): binomial, beta-binomial, Poisson, negative binomial, geometric, zeta, normal, log normal, inverse Gauss, logistic, Laplace, Cauchy, Student t, exponential, Pareto, gamma, and Weibull to frequency (histogram) data, possibly plotting the frequency polygon of fitted values with the histogram.

**Usage**

```

fit.dist(
  y,
  ni,
  distribution = "normal",
  breaks = FALSE,
  delta = 1,
  censor = FALSE,
  exact = TRUE,
  plot = FALSE,
  add = FALSE,
  xlab = deparse(substitute(y)),
  ylab = "Probability",

```

```
xlim = range(y),
main = paste("Histogram of", deparse(substitute(y))),
...
)
```

### Arguments

y	Vector of observations.
ni	Corresponding vector of frequencies.
distribution	Character string specifying the distribution.
breaks	If TRUE, y contains breaks between categories instead of mid-points.
delta	Scalar or vector giving the unit of measurement (always one for discrete data) for each response value, set to unity by default. For example, if a response is measured to two decimals, delta=0.01.
censor	If TRUE, the last category is right censored.
exact	If FALSE, uses the approximations for certain distributions in Lindsey (1995).
plot	If TRUE, plots the histogram of observed frequencies and the frequency polygon of fitted values.
add	If TRUE, adds a new frequency polygon of fitted values without replotting the histogram.
xlab	Plotting control options.
ylab	Plotting control options.
xlim	Plotting control options.
main	Plotting control options.
...	Plotting control options.

### Author(s)

J.K. Lindsey

### References

Lindsey, J.K. (1995) Introductory Statistics: A Modelling Approach. Oxford: Oxford University Press.

### Examples

```
f <- c(215, 1485, 5331, 10649, 14959, 11929, 6678, 2092, 342)
y <- seq(0,8)
fit.dist(y, f, "binomial", plot=TRUE, xlab="Number of males",
main="Distribution of males in families of 8 children")
#
f <- c(1,1,6,3,4,3,9,6,5,16,4,11,6,11,3,4,5,6,4,4,5,1,1,4,1,2,
0,2,0,0,1)
y <- seq(1100,4100,by=100)
fit.dist(y, f, "normal", delta=100, plot=TRUE,
```

```

xlab="Monthly salary (dollars)",
main="Distribution of women mathematicians' salaries")
fit.dist(y, f, "log normal", delta=100, plot=TRUE, add=TRUE, lty=3)
fit.dist(y, f, "logistic", delta=100, exact=FALSE, plot=TRUE, add=TRUE, lty=2)

```

fmr

*Generalized Nonlinear Regression Models with Two or Three Point Mixtures*

## Description

fmr fits user specified nonlinear regression equations to the location parameter of the common one and two parameter distributions. (The log of the scale parameter is estimated to ensure positivity.)

## Usage

```

fmr(
  y = NULL,
  distribution = "normal",
  mu = NULL,
  mix = NULL,
  linear = NULL,
  pmu = NULL,
  pmix = NULL,
  pshape = NULL,
  censor = "right",
  exact = FALSE,
  wt = 1,
  delta = 1,
  common = FALSE,
  envir = parent.frame(),
  print.level = 0,
  tysize = abs(p),
  ndigit = 10,
  gradtol = 1e-05,
  stepmax = 10 * sqrt(p %>% p),
  steptol = 1e-05,
  iterlim = 100,
  fscale = 1
)

```

## Arguments

**y** A response vector for uncensored data, a two column matrix for binomial data or censored data, with the second column being the censoring indicator (1: uncensored, 0: right censored, -1: left censored), or an object of class, response (created by [restovec](#)) or repeated (created by [rmna](#) or [lvna](#)). If the repeated

	data object contains more than one response variable, give that object in <code>envir</code> and give the name of the response variable to be used here.
<code>distribution</code>	Either a character string containing the name of the distribution or a function giving the <code>-log</code> likelihood and calling the location and mixture functions. Distributions are binomial, beta binomial, double binomial, multiplicative binomial, Poisson, negative binomial, double Poisson, multiplicative Poisson, gamma count, Consul, geometric, normal, inverse Gauss, logistic, exponential, gamma, Weibull, extreme value, Pareto, Cauchy, Student t, Laplace, and Levy. (For definitions of distributions, see the corresponding <code>[dpqr]</code> distribution help.)
<code>mu</code>	A user-specified function of <code>pmu</code> , and possibly <code>linear</code> , giving the regression equation for the location. This may contain a linear part as the second argument to the function. It may also be a formula beginning with <code>~</code> , specifying either a linear regression function for the location parameter in the Wilkinson and Rogers notation or a general function with named unknown parameters. If it contains unknown parameters, the keyword <code>linear</code> may be used to specify a linear part. If nothing is supplied, the location is taken to be constant unless the linear argument is given.
<code>mix</code>	A user-specified function of <code>pmix</code> , and possibly <code>linear</code> , giving the regression equation for the mixture parameter. This may contain a linear part as the second argument to the function. It may also be a formula beginning with <code>~</code> , specifying either a linear regression function for the mixture parameter in the Wilkinson and Rogers notation or a general function with named unknown parameters. If it contains unknown parameters, the keyword <code>linear</code> may be used to specify a linear part. If nothing is supplied, this parameter is taken to be constant. This parameter is the logit of the mixture probability.
<code>linear</code>	A formula beginning with <code>~</code> in W&R notation, or list of two such expressions, specifying the linear part of the regression function for the location or location and mixture parameters.
<code>pmu</code>	Vector of initial estimates for the location parameters. If <code>mu</code> is a formula with unknown parameters, their estimates must be supplied either in their order of appearance in the expression or in a named list.
<code>pmix</code>	Vector of initial estimates for the mixture parameters. If <code>mix</code> is a formula with unknown parameters, their estimates must be supplied either in their order of appearance in the expression or in a named list.
<code>pshape</code>	An initial estimate for the shape parameter.
<code>cursor</code>	<code>right</code> , <code>left</code> , or <code>both</code> indicating where the mixing distribution is placed. <code>both</code> is only possible for binomial data.
<code>exact</code>	If <code>TRUE</code> , fits the exact likelihood function for continuous data by integration over intervals of observation given in <code>delta</code> , i.e. interval censoring.
<code>wt</code>	Weight vector.
<code>delta</code>	Scalar or vector giving the unit of measurement (always one for discrete data) for each response value, set to unity by default - for example, if a response is measured to two decimals, <code>delta=0.01</code> . If the response is transformed, this must be multiplied by the Jacobian. The transformation cannot contain unknown parameters. For example, with a log transformation, <code>delta=1/y</code> .

<code>common</code>	If TRUE, <code>mu</code> and <code>mix</code> must both be either functions with, as argument, a vector of parameters having some or all elements in common between them so that indexing is in common between them or formulae with unknowns. All parameter estimates must be supplied in <code>pmu</code> . If FALSE, parameters are distinct between the two functions and indexing starts at one in each function.
<code>envir</code>	Environment in which model formulae are to be interpreted or a data object of class, <code>repeated</code> , <code>tccov</code> , or <code>tvcov</code> ; the name of the response variable should be given in <code>y</code> . If <code>y</code> has class <code>repeated</code> , it is used as the environment.
<code>print.level</code>	Arguments controlling <a href="#">nlm</a> .
<code>tysize</code>	Arguments controlling <a href="#">nlm</a> .
<code>ndigit</code>	Arguments controlling <a href="#">nlm</a> .
<code>gradtol</code>	Arguments controlling <a href="#">nlm</a> .
<code>stepmax</code>	Arguments controlling <a href="#">nlm</a> .
<code>steptol</code>	Arguments controlling <a href="#">nlm</a> .
<code>iterlim</code>	Arguments controlling <a href="#">nlm</a> .
<code>fscale</code>	Arguments controlling <a href="#">nlm</a> .

### Details

For the Poisson and related distributions, the mixture involves the zero category. For the binomial and related distributions, it involves the two extreme categories. For all other distributions, it involves either left or right censored individuals. A user-specified `-log` likelihood can also be supplied for the distribution.

Nonlinear regression models can be supplied as formulae where parameters are unknowns in which case factor variables cannot be used and parameters must be scalars. (See [finterp](#).)

The printed output includes the `-log` likelihood (not the deviance), the corresponding AIC, the maximum likelihood estimates, standard errors, and correlations.

### Value

A list of class `gnlm` is returned that contains all of the relevant information calculated, including error codes.

### Author(s)

J.K. Lindsey

### See Also

[finterp](#), [glm](#), [gnlr](#), [gnlr3](#), [lm](#).



## Examples

```

sex <- c(rep(0,10),rep(1,10))
sexf <- gl(2,10)
age <- c(8,10,12,12,8,7,16,7,9,11,8,9,14,12,12,11,7,7,7,12)
y <- cbind(c(9.2, 7.3,13.0, 6.9, 3.9,14.9,17.8, 4.8, 6.4, 3.3,17.2,
14.4,17.0, 5.0,17.3, 3.8,19.4, 5.0, 2.0,19.0),
c(0,1,0,1,1,1,0,1,0,1,1,1,1,1,1,1,1,1,1,1))
# y <- cbind(rweibull(20,2,2+2*sex+age),rbinom(20,1,0.7))
# log linear regression with Weibull distribution with a point mass
# for right censored individuals
mu <- function(p) exp(p[1]+p[2]*sex+p[3]*age)
fmr(y, dist="Weibull", mu=mu, pmu=c(4,0,0), pmix=0.5, pshape=1)
# or equivalently
fmr(y, dist="Weibull", mu=function(p,linear) exp(linear),
linear=~sexf+age, pmu=c(4,0,0), pmix=0.5, pshape=1)
# or
fmr(y, dist="Weibull", mu=~exp(b0+b1*sex+b2*age), pmu=list(b0=4,b1=0,b2=0),
pmix=0.5, pshape=1)
#
# include logistic regression for the mixture parameter
mix <- function(p) p[1]+p[2]*sex
fmr(y, dist="Weibull", mu=~exp(a+b*age), mix=mix, pmu=c(4,0),
pmix=c(10,0), pshape=0.5)
# or equivalently
fmr(y, dist="Weibull", mu=function(p,linear) exp(linear),
linear=list(~age,~sexf), pmu=c(4,0), pmix=c(10,0), pshape=0.5)
# or
fmr(y, dist="Weibull", mu=~exp(b0+b1*age), mix=~c0+c1*sex,
pmu=list(b0=4,b1=0), pmix=list(c0=10,c1=0), pshape=0.5)
#
# generate zero-inflated negative binomial data
x1 <- rpois(50,4)
x2 <- rpois(50,4)
ind <- rbinom(50,1,1/(1+exp(-1-0.1*x1)))
y <- ifelse(ind,rnbinom(50,3,mu=exp(1+0.2*x2)),0)
# standard Poisson models
gnlr(y, dist="Poisson", mu=~exp(a), pmu=1)
gnlr(y, dist="Poisson", mu=~exp(linear), linear=~x2, pmu=c(1,0.2))
# zero-inflated Poisson ZIP
fmr(y, dist="Poisson", mu=~exp(a), pmu=1, pmix=0)
fmr(y, dist="Poisson", mu=~exp(linear), linear=~x2, pmu=c(1,0.2), pmix=0)
fmr(y, dist="Poisson", mu=~exp(a), mix=~x1, pmu=1, pmix=c(1,0))
fmr(y, dist="Poisson", mu=~exp(linear), linear=~x2, mix=~x1, pmu=c(1,0.2),
pmix=c(1,0))
# zero-inflated negative binomial
fmr(y, dist="negative binomial", mu=~exp(a), pmu=1, pshape=0, pmix=0)
fmr(y, dist="negative binomial", mu=~exp(linear), linear=~x2, pmu=c(1,0.2),
pshape=0, pmix=0)
fmr(y, dist="negative binomial", mu=~exp(a), mix=~x1, pmu=1, pshape=0,
pmix=c(1,0))
fmr(y, dist="negative binomial", mu=~exp(linear), linear=~x2, mix=~x1,
pmu=c(1,0.2), pshape=0, pmix=c(1,0))

```

---

gnlr	<i>Generalized Nonlinear Regression Models for One and Two Parameter Distributions</i>
------	--

---

### Description

gnlr fits user-specified nonlinear regression equations to one or both parameters of the common one and two parameter distributions. A user-specified -log likelihood can also be supplied for the distribution. Most distributions allow data to be left, right, and/or interval censored.

### Usage

```
gnlr(
  y = NULL,
  distribution = "normal",
  pmu = NULL,
  pshape = NULL,
  mu = NULL,
  shape = NULL,
  linear = NULL,
  exact = FALSE,
  wt = 1,
  delta = 1,
  shfn = FALSE,
  common = FALSE,
  envir = parent.frame(),
  print.level = 0,
  typsize = abs(p),
  ndigit = 10,
  gradtol = 1e-05,
  stepmax = 10 * sqrt(p %% p),
  steptol = 1e-05,
  iterlim = 100,
  fscale = 1
)
```

### Arguments

**y** A response vector for uncensored data, a two column matrix for binomial data or censored data, with the second column being the censoring indicator (1: uncensored, 0: right censored, -1: left censored), or an object of class, response (created by [restovec](#)) or repeated (created by [rmna](#) or [lvna](#)). If the repeated data object contains more than one response variable, give that object in `envir` and give the name of the response variable to be used here. The beta, simplex, and two-sided power distributions for proportions do not allow left or right censoring (only interval censoring).

distribution	<p>Either a character string containing the name of the distribution or a function giving the <math>-\log</math> likelihood. (In the latter case, all initial parameter estimates are supplied in <code>pmu</code>.)</p> <p>Distributions are binomial, beta binomial, double binomial, mult(iplicative) binomial, Poisson, negative binomial, double Poisson, mult(iplicative) Poisson, gamma count, Consul generalized Poisson, logarithmic series, geometric, normal, inverse Gauss, logistic, exponential, gamma, Weibull, extreme value, Cauchy, Pareto, Laplace, Levy, beta, simplex, and two-sided power. All but the binomial-based distributions and the beta, simplex, and two-sided power distributions may be right and/or left censored. (For definitions of distributions, see the corresponding [dpqr]distribution help.)</p>
pmu	<p>Vector of initial estimates for the location parameters. If <code>mu</code> is a formula with unknown parameters, their estimates must be supplied either in their order of appearance in the expression or in a named list. If <code>distribution</code> is a user-supplied <math>-\log</math> likelihood function, all initial estimates must be supplied here.</p>
pshape	<p>Vector of initial estimates for the shape parameters. If <code>shape</code> is a formula with unknown parameters, their estimates must be supplied either in their order of appearance in the expression or in a named list.</p>
mu	<p>A user-specified function of <code>pmu</code>, and possibly <code>linear</code>, giving the regression equation for the location. This may contain a linear part as the second argument to the function. It may also be a formula beginning with <code>~</code>, specifying either a linear regression function for the location parameter in the Wilkinson and Rogers notation or a general function with named unknown parameters. If it contains unknown parameters, the keyword <code>linear</code> may be used to specify a linear part. If nothing is supplied, the location is taken to be constant unless the <code>linear</code> argument is given.</p>
shape	<p>A user-specified function of <code>pshape</code>, and possibly <code>linear</code> and/or <code>mu</code>, giving the regression equation for the dispersion or shape parameter. This may contain a linear part as the second argument to the function and the location function as last argument (in which case <code>shfn</code> must be set to <code>TRUE</code>). It may also be a formula beginning with <code>~</code>, specifying either a linear regression function for the shape parameter in the Wilkinson and Rogers notation or a general function with named unknown parameters. If it contains unknown parameters, the keyword <code>linear</code> may be used to specify a linear part and the keyword <code>mu</code> to specify a function of the location parameter. If nothing is supplied, this parameter is taken to be constant unless the <code>linear</code> argument is given. This parameter is the logarithm of the usual one.</p>
linear	<p>A formula beginning with <code>~</code> in W&amp;R notation, specifying the linear part of the regression function for the location parameter or list of two such expressions for the location and/or shape parameters.</p>
exact	<p>If <code>TRUE</code>, fits the exact likelihood function for continuous data by integration over intervals of observation given in <code>delta</code>, i.e. interval censoring.</p>
wt	<p>Weight vector.</p>
delta	<p>Scalar or vector giving the unit of measurement (always one for discrete data) for each response value, set to unity by default. For example, if a response is measured to two decimals, <code>delta=0.01</code>. If the response is transformed, this</p>

	must be multiplied by the Jacobian. The transformation cannot contain unknown parameters. For example, with a log transformation, $\delta = 1/y$ . (The delta values for the censored response are ignored.)
shfn	If true, the supplied shape function depends on the location (function). The name of this location function must be the last argument of the shape function.
common	If TRUE, mu and shape must both be either functions with, as argument, a vector of parameters having some or all elements in common between them so that indexing is in common between them or formulae with unknowns. All parameter estimates must be supplied in pmu. If FALSE, parameters are distinct between the two functions and indexing starts at one in each function.
envir	Environment in which model formulae are to be interpreted or a data object of class, repeated, tccov, or tvcov; the name of the response variable should be given in y. If y has class repeated, it is used as the environment.
print.level	Arguments controlling <a href="#">nlm</a> .
typsize	Arguments controlling <a href="#">nlm</a> .
ndigit	Arguments controlling <a href="#">nlm</a> .
gradtol	Arguments controlling <a href="#">nlm</a> .
stepmax	Arguments controlling <a href="#">nlm</a> .
steptol	Arguments controlling <a href="#">nlm</a> .
iterlim	Arguments controlling <a href="#">nlm</a> .
fscale	Arguments controlling <a href="#">nlm</a> .

### Details

Nonlinear regression models can be supplied as formulae where parameters are unknowns in which case factor variables cannot be used and parameters must be scalars. (See [finterp](#).)

The printed output includes the -log likelihood (not the deviance), the corresponding AIC (which needs to be multiplied by 2 to match 'stats::glm'), the maximum likelihood estimates, standard errors, and correlations.

### Value

A list of class `gnlm` is returned that contains all of the relevant information calculated, including error codes.

### Author(s)

J.K. Lindsey

### See Also

[finterp](#), [fmr](#), [glm](#), [gnlmix](#), [glmm](#), [gnlmm](#), [gnlr3](#), [lm](#), [nlr](#), [nls](#).

**Examples**

```

sex <- c(rep(0,10),rep(1,10))
sexf <- gl(2,10)
age <- c(8,10,12,12,8,7,16,7,9,11,8,9,14,12,12,11,7,7,7,12)
y <- cbind(c(9.2, 7.3,13.0, 6.9, 3.9,14.9,17.8, 4.8, 6.4, 3.3,17.2,
14.4,17.0, 5.0,17.3, 3.8,19.4, 5.0, 2.0,19.0),
c(0,1,0,1,1,1,0,1,0,1,1,1,1,1,1,1,1,1,1,1))
# y <- cbind(rweibull(20,2,2+2*sex+age),rbinom(20,1,0.7))
# linear regression with inverse Gauss distribution
mu <- function(p) p[1]+p[2]*sex+p[3]*age
gnlr(y, dist="inverse Gauss", mu=mu, pmu=c(3,0,0), pshape=1)
# or equivalently
gnlr(y, dist="inverse Gauss", mu=~sexf+age, pmu=c(3,0,0), pshape=1)
# or
gnlr(y, dist="inverse Gauss", linear=~sexf+age, pmu=c(3,0,0), pshape=1)
# or
gnlr(y, dist="inverse Gauss", mu=~b0+b1*sex+b2*age,
pmu=list(b0=3,b1=0,b2=0), pshape=1)
#
# nonlinear regression with inverse Gauss distribution
mu <- function(p, linear) exp(linear)
gnlr(y, dist="inverse Gauss", mu=mu, linear=~sexf+age, pmu=c(3,0,0),
pshape=-1)
# or equivalently
gnlr(y, dist="inverse Gauss", mu=~exp(b0+b1*sex+b2*age),
pmu=list(b0=3,b1=0,b2=0), pshape=-1)
# or
gnlr(y, dist="inverse Gauss", mu=~exp(linear), linear=~sexf+age,
pmu=c(3,0,0), pshape=-1)
#
# include regression for the shape parameter with same mu function
shape <- function(p) p[1]+p[2]*sex+p[3]*age
gnlr(y, dist="inverse Gauss", mu=mu, linear=~sexf+age, shape=shape,
pmu=c(3,0,0), pshape=c(3,0,0))
# or equivalently
gnlr(y, dist="inverse Gauss", mu=mu, linear=~sexf+age,
shape=~sexf+age, pmu=c(3,0,0), pshape=c(3,0,0))
# or
gnlr(y, dist="inverse Gauss", mu=mu, linear=list(~sexf+age,~sexf+age),
pmu=c(3,0,0),pshape=c(3,0,0))
# or
gnlr(y, dist="inverse Gauss", mu=mu, linear=~sexf+age,
shape=~c0+c1*sex+c2*age, pmu=c(3,0,0),
pshape=list(c0=3,c1=0,c2=0))
#
# shape as a function of the location
shape <- function(p, mu) p[1]+p[2]*sex+p[3]*mu
gnlr(y, dist="inverse Gauss", mu=~age, shape=shape, pmu=c(3,0),
pshape=c(3,0,0), shfn=TRUE)
# or
gnlr(y, dist="inverse Gauss", mu=~age, shape=~a+b*sex+c*mu, pmu=c(3,0),
pshape=c(3,0,0), shfn=TRUE)

```

```

#
# common parameter in location and shape functions for age
mu <- function(p) exp(p[1]+p[2]*age)
shape <- function(p, mu) p[3]+p[4]*sex+p[2]*age
gnlr(y, dist="inverse Gauss", mu=mu, shape=shape, pmu=c(3,0,1,0),
common=TRUE)
# or
gnlr(y, dist="inverse Gauss", mu=~exp(a+b*age), shape=~c+d*sex+b*age,
pmu=c(3,0,1,0), common=TRUE)
#
# user-supplied -log likelihood function
y <- rnorm(20,2+3*sex,2)
dist <- function(p)-sum(dnorm(y,p[1]+p[2]*sex,p[3],log=TRUE))
gnlr(y, dist=dist,pmu=1:3)
dist <- ~-sum(dnorm(y,a+b*sex,v,log=TRUE))
gnlr(y, dist=dist,pmu=1:3)

```

gnlr3

*Generalized Nonlinear Regression Models for Three Parameter Distributions*

## Description

gnlr3 fits user specified nonlinear regression equations to one, two, or all three parameters of three parameter distributions. Continuous data may be left, right, and/or interval censored.

## Usage

```

gnlr3(
  y = NULL,
  distribution = "normal",
  mu = NULL,
  shape = NULL,
  family = NULL,
  linear = NULL,
  pmu = NULL,
  pshape = NULL,
  pfamily = NULL,
  exact = FALSE,
  wt = 1,
  common = FALSE,
  delta = 1,
  envir = parent.frame(),
  print.level = 0,
  tysize = abs(p),
  ndigit = 10,
  gradtol = 1e-05,

```

```

    stepmax = 10 * sqrt(p %*% p),
    steptol = 1e-05,
    iterlim = 100,
    fscale = 1
)

```

## Arguments

<code>y</code>	The response vector for uncensored data, two columns for censored data, with the second being the censoring indicator (1: uncensored, 0: right censored, -1: left censored.), or an object of class, <code>response</code> (created by <code>restovec</code> ) or <code>repeated</code> (created by <code>rmna</code> or <code>lvna</code> ). If the <code>repeated</code> data object contains more than one response variable, give that object in <code>envir</code> and give the name of the response variable to be used here.
<code>distribution</code>	Either a character string containing the name of the distribution or a function giving the <code>-log</code> likelihood and calling the location, shape, and family functions. Distributions are Box-Cox transformed normal, generalized inverse Gauss, generalized logistic, Hjorth, generalized gamma, Burr, generalized Weibull, power exponential, Student t, generalized extreme value, power variance function Poisson, and skew Laplace. (For definitions of distributions, see the corresponding <code>[dpqr]distribution help</code> .)
<code>mu</code>	A user-specified function of <code>pmu</code> , and possibly <code>linear</code> , giving the regression equation for the location. This may contain a linear part as the second argument to the function. It may also be a formula beginning with <code>~</code> , specifying either a linear regression function for the location parameter in the Wilkinson and Rogers notation or a general function with named unknown parameters. If it contains unknown parameters, the keyword <code>linear</code> may be used to specify a linear part. If nothing is supplied, the location is taken to be constant unless the <code>linear</code> argument is given.
<code>shape</code>	A user-specified function of <code>pshape</code> , and possibly <code>linear</code> , giving the regression equation for the dispersion or shape parameter. This may contain a linear part as the second argument to the function. It may also be a formula beginning with <code>~</code> , specifying either a linear regression function for the shape parameter in the Wilkinson and Rogers notation or a general function with named unknown parameters. If it contains unknown parameters, the keyword <code>linear</code> may be used to specify a linear part. If nothing is supplied, this parameter is taken to be constant unless the <code>linear</code> argument is given. This parameter is the logarithm of the usual one.
<code>family</code>	A user-specified function of <code>pfamily</code> , and possibly <code>linear</code> , for the regression equation of the third (family) parameter of the distribution. This may contain a linear part that is the second argument to the function. It may also be a formula beginning with <code>~</code> , specifying either a linear regression function for the family parameter in the Wilkinson and Rogers notation or a general function with named unknown parameters. If neither is supplied, this parameter is taken to be constant unless the <code>linear</code> argument is given. In most cases, this parameter is the logarithm of the usual one.
<code>linear</code>	A formula beginning with <code>~</code> in W&R notation, specifying the linear part of the

	regression function for the location parameters or list of three such expressions for the location, shape, and/or family parameters.
pmu	Vector of initial estimates for the location parameters. If <code>mu</code> is a formula with unknown parameters, their estimates must be supplied either in their order of appearance in the expression or in a named list.
pshape	Vector of initial estimates for the shape parameters. If <code>shape</code> is a formula with unknown parameters, their estimates must be supplied either in their order of appearance in the expression or in a named list.
pfamily	Vector of initial estimates for the family parameters. If <code>family</code> is a formula with unknown parameters, their estimates must be supplied either in their order of appearance in the expression or in a named list.
exact	If TRUE, fits the exact likelihood function for continuous data by integration over intervals of observation given in <code>delta</code> , i.e. interval censoring.
wt	Weight vector.
common	If TRUE, at least two of <code>mu</code> , <code>shape</code> , and <code>family</code> must both be either functions with, as argument, a vector of parameters having some or all elements in common between them so that indexing is in common between them or formulae with unknowns. All parameter estimates must be supplied in <code>pmu</code> . If FALSE, parameters are distinct between the two functions and indexing starts at one in each function.
delta	Scalar or vector giving the unit of measurement (always one for discrete data) for each response value, set to unity by default - for example, if a response is measured to two decimals, <code>delta=0.01</code> . If the response is transformed, this must be multiplied by the Jacobian. The transformation cannot contain unknown parameters. For example, with a log transformation, <code>delta=1/y</code> . (The delta values for the censored response are ignored.)
envir	Environment in which model formulae are to be interpreted or a data object of class, <code>repeated</code> , <code>tccov</code> , or <code>tvcov</code> ; the name of the response variable should be given in <code>y</code> . If <code>y</code> has class <code>repeated</code> , it is used as the environment.
print.level	Arguments controlling <a href="#">nlm</a> .
typsize	Arguments controlling <a href="#">nlm</a> .
ndigit	Arguments controlling <a href="#">nlm</a> .
gradtol	Arguments controlling <a href="#">nlm</a> .
stepmax	Arguments controlling <a href="#">nlm</a> .
steptol	Arguments controlling <a href="#">nlm</a> .
iterlim	Arguments controlling <a href="#">nlm</a> .
fscale	Arguments controlling <a href="#">nlm</a> .

### Details

Nonlinear regression models can be supplied as formulae where parameters are unknowns in which case factor variables cannot be used and parameters must be scalars. (See [finterp](#).)

The printed output includes the -log likelihood (not the deviance), the corresponding AIC, the maximum likelihood estimates, standard errors, and correlations.



**Value**

A list of class `gnlm` is returned that contains all of the relevant information calculated, including error codes.

**Author(s)**

J.K. Lindsey

**See Also**

[finterp](#), [fmr](#), [glm](#), [gnlr](#), [lm](#), [nlr](#), [nls](#).

**Examples**

```
sex <- c(rep(0,10),rep(1,10))
sexf <- gl(2,10)
age <- c(8,10,12,12,8,7,16,7,9,11,8,9,14,12,12,11,7,7,7,12)
y <- cbind(c(9.2, 7.3,13.0, 6.9, 3.9,14.9,17.8, 4.8, 6.4, 3.3,17.2,
14.4,17.0, 5.0,17.3, 3.8,19.4, 5.0, 2.0,19.0),
c(0,1,0,1,1,1,0,1,0,1,1,1,1,1,1,1,1,1,1,1))
# y <- cbind(rweibull(20,2,2+2*sex+age),rbinom(20,1,0.7))
# log linear regression with the generalized Weibull distribution
mu <- function(p) exp(p[1]+p[2]*sex+p[3]*age)
gnlr3(y, dist="Weibull", mu=mu, pmu=c(3,1,0), pshape=2, pfamily=-2)
# or equivalently
mu1 <- function(p,linear) exp(linear)
gnlr3(y, dist="Weibull", mu=mu1, linear=~sex+age, pmu=c(3,1,0),
pshape=2, pfamily=-2)
# or
gnlr3(y, dist="Weibull", mu=~exp(b0+b1*sex+b2*age),
pmu=list(b0=3,b1=1,b2=0), pshape=2, pfamily=-2)
#
# include regression for the shape parameter with same mu function
shape <- function(p) p[1]+p[2]*sex+p[3]*age
gnlr3(y, dist="Weibull", mu=mu, shape=shape,
pmu=c(3,1,0), pshape=c(2,0,0), pfamily=-2)
# or equivalently
gnlr3(y, dist="Weibull", mu=mu1, linear=list(~sexf+age,~sex+age,NULL),
pmu=c(3,1,0), pshape=c(2,0,0), pfamily=-2)
# or
gnlr3(y, dist="Weibull", mu=~exp(b0+b1*sex+b2*age),
shape=~c0+c1*sex+c2*age, pmu=c(3,1,0),
pshape=list(c0=2,c1=0,c2=0), pfamily=-2)
# include regression for the family parameter with same mu
# and shape functions
family <- function(p) p[1]+p[2]*sex+p[3]*age
gnlr3(y, dist="Weibull", mu=mu1, linear=~sexf+age, shape=shape,
family=family, pmu=c(2.5,1,0), pshape=c(2,0,0), pfamily=c(-2,0,0))
# or equivalently
gnlr3(y, dist="Weibull", mu=mu1, linear=list(~sex+age,~sex+age,~sex+age),
pmu=c(2.5,1,0), pshape=c(2,0,0), pfamily=c(-2,0,0))
# or
```

```

gnlr3(y, dist="Weibull", mu=~exp(b0+b1*sex+b2*age),
shape=~c0+c1*sex+c2*age, family=~d0+d1*sex+d2*age,
pmu=list(b0=2.5,b1=1,b2=0), pshape=list(c0=2,c1=0,c2=0),
pfamily=list(d0=-2,d1=0,d2=0))
#
# common parameters
mu <- function(p) exp(p[1]+p[2]*sex+p[3]*age)
shape <- function(p) p[4]+p[5]*sex+p[3]*age
family <- function(p) p[6]+p[7]*sex+p[3]*age
gnlr3(y, dist="Weibull", mu=mu, shape=shape, family=family,
pmu=c(2.5,1,0,1,0,1,0), common=TRUE)
# or
gnlr3(y, dist="Weibull", mu=~exp(a+b*sex+c*age), shape=~d+e*sex+c*age,
family=~f+g*sex+c*age, pmu=c(2.5,1,0,1,0,1,0), common=TRUE)

```

nlr

*Nonlinear Normal, Gamma, and Inverse Gaussian Regression Models***Description**

nlr fits a user-specified nonlinear regression equation by least squares (normal) or its generalization for the gamma and inverse Gauss distributions.

**Usage**

```

nlr(
  y = NULL,
  mu = NULL,
  pmu = NULL,
  distribution = "normal",
  wt = 1,
  delta = 1,
  envir = parent.frame(),
  print.level = 0,
  tysize = abs(pmu),
  ndigit = 10,
  gradtol = 1e-05,
  stepmax = 10 * sqrt(pmu %*% pmu),
  steptol = 1e-05,
  iterlim = 100,
  fscale = 1
)

```

**Arguments**

y The response vector or an object of class, response (created by [restovec](#)) or repeated (created by [rmna](#) or [lvna](#)).

mu	A function of p giving the regression equation for the mean or a formula beginning with ~, specifying either a linear regression function in the Wilkinson and Rogers notation or a general nonlinear function with named unknown parameters.
pmu	Vector of initial estimates of the parameters. If mu is a formula with unknown parameters, their estimates must be supplied either in their order of appearance in the expression or in a named list.
distribution	The distribution to be used: normal, gamma, or inverse Gauss.
wt	Weight vector.
delta	Scalar or vector giving the unit of measurement for each response value, set to unity by default. For example, if a response is measured to two decimals, delta=0.01. If the response is transformed, this must be multiplied by the Jacobian. For example, with a log transformation, delta=1/y.
envir	Environment in which model formulae are to be interpreted or a data object of class, repeated, tccov, or tvcov. If y has class repeated, it is used as the environment.
print.level	Arguments controlling <a href="#">nlm</a> .
typsize	Arguments controlling <a href="#">nlm</a> .
ndigit	Arguments controlling <a href="#">nlm</a> .
gradtol	Arguments controlling <a href="#">nlm</a> .
stepmax	Arguments controlling <a href="#">nlm</a> .
steptol	Arguments controlling <a href="#">nlm</a> .
iterlim	Arguments controlling <a href="#">nlm</a> .
fscale	Arguments controlling <a href="#">nlm</a> .

### Details

A nonlinear regression model can be supplied as a formula where parameters are unknowns in which case factor variables cannot be used and parameters must be scalars. (See [finterp](#).)

The printed output includes the -log likelihood (not the deviance), the corresponding AIC, the parameter estimates, standard errors, and correlations.

### Value

A list of class nlr is returned that contains all of the relevant information calculated, including error codes.

### Author(s)

J.K. Lindsey

### See Also

[finterp](#), [fmr](#), [glm](#), [glmm](#), [gnlmm](#), [gnlr](#), [gnlr3](#), [lm](#), [nls](#).

**Examples**

```

x <- c(3,5,0,0,0,3,2,2,2,7,4,0,0,2,2,2,0,1,3,4)
y <- c(5.8,11.6,2.2,2.7,2.3,9.4,11.7,3.3,1.5,14.6,9.6,7.4,10.7,6.9,
2.6,17.3,2.8,1.2,1.0,3.6)
# rgamma(20,2,scale=0.2+2*exp(0.1*x))
# linear least squares regression
mu1 <- function(p) p[1]+p[2]*x
summary(lm(y~x))
nlr(y, mu=mu1, pmu=c(3,0))
# or
nlr(y, mu=~x, pmu=c(3,0))
# or
nlr(y, mu=~b0+b1*x, pmu=c(3,0))
# linear gamma regression
nlr(y, dist="gamma", mu=~x, pmu=c(3,0))
# nonlinear regression
mu2 <- function(p) p[1]+p[2]*exp(p[3]*x)
nlr(y, mu=mu2, pmu=c(0.2,3,0.2))
# or
nlr(y, mu=~b0+c0*exp(c1*x), pmu=list(b0=0.2,c0=3,c1=0.2))
# with gamma distribution
nlr(y, dist="gamma", mu=~b0+c0*exp(c1*x), pmu=list(b0=0.2,c0=3,c1=0.2))

```

---

nordr

*Nonlinear Ordinal Regression Models*


---

**Description**

nordr fits arbitrary nonlinear regression functions (with logistic link) to ordinal response data by proportional odds, continuation ratio, or adjacent categories.

**Usage**

```

nordr(
  y = NULL,
  distribution = "proportional",
  mu = NULL,
  linear = NULL,
  pmu = NULL,
  pintercept = NULL,
  weights = NULL,
  envir = parent.frame(),
  print.level = 0,
  ndigit = 10,
  gradtol = 1e-05,
  steptol = 1e-05,
  fscale = 1,

```

```

iterlim = 100,
tysize = abs(p),
stepmax = 10 * sqrt(p %% p)
)

```

## Arguments

y	A vector of ordinal responses, integers numbered from zero to one less than the number of categories or an object of class, response (created by <code>restovec</code> ) or repeated (created by <code>rmna</code> ) or <code>lvna</code> ). If the repeated data object contains more than one response variable, give that object in <code>envir</code> and give the name of the response variable to be used here.
distribution	The ordinal distribution: proportional odds, continuation ratio, or adjacent categories.
mu	User-specified function of <code>pmu</code> , and possibly linear, giving the logistic regression equation. This must contain the first intercept. It may contain a linear part as the second argument to the function. It may also be a formula beginning with <code>~</code> , specifying a logistic regression function for the location parameter, either a linear one using the Wilkinson and Rogers notation or a general function with named unknown parameters. If it contains unknown parameters, the keyword <code>linear</code> may be used to specify a linear part. If nothing is supplied, the location is taken to be constant unless the linear argument is given.
linear	A formula beginning with <code>~</code> in W&R notation, specifying the linear part of the logistic regression function.
pmu	Vector of initial estimates for the regression parameters, including the first intercept. If <code>mu</code> is a formula with unknown parameters, their estimates must be supplied either in their order of appearance in the expression or in a named list.
pintercept	Vector of initial estimates for the contrasts with the first intercept parameter (difference in intercept for successive categories): two less than the number of different ordinal values.
weights	Weight vector for use with contingency tables.
envir	Environment in which model formulae are to be interpreted or a data object of class, repeated, <code>tccov</code> , or <code>tvcov</code> ; the name of the response variable should be given in <code>y</code> . If <code>y</code> has class repeated, it is used as the environment.
print.level	Arguments controlling <a href="#">nlm</a> .
ndigit	Arguments controlling <a href="#">nlm</a> .
gradtol	Arguments controlling <a href="#">nlm</a> .
steptol	Arguments controlling <a href="#">nlm</a> .
fscale	Arguments controlling <a href="#">nlm</a> .
iterlim	Arguments controlling <a href="#">nlm</a> .
tysize	Arguments controlling <a href="#">nlm</a> .
stepmax	Arguments controlling <a href="#">nlm</a> .

**Details**

Nonlinear regression models can be supplied as formulae where parameters are unknowns in which case factor variables cannot be used and parameters must be scalars. (See [finterp](#).)

The printed output includes the -log likelihood (not the deviance), the corresponding AIC, the maximum likelihood estimates, standard errors, and correlations.

**Value**

A list of class nordr is returned that contains all of the relevant information calculated, including error codes.

**Author(s)**

J.K. Lindsey

**See Also**

[finterp](#), [fmr](#), [glm](#), [glmm](#), [gnlmm](#), [gnlr](#), [gnlr3](#), [nlr](#), [ordglm](#)

**Examples**

```
# McCullagh (1980) JRSS B42, 109-142
# tonsil size: 2x3 contingency table
y <- c(0:2,0:2)
carrier <- c(rep(0,3),rep(1,3))
carrierf <- gl(2,3,6)
wt <- c(19,29,24,
497,560,269)
pmu <- c(-1,0.5)
mu <- function(p) c(rep(p[1],3),rep(p[1]+p[2],3))
# proportional odds
# with mean function
nordr(y, dist="prop", mu=mu, pmu=pmu, weights=wt, pintercept=1.5)
# using Wilkinson and Rogers notation
nordr(y, dist="prop", mu=~carrierf, pmu=pmu, weights=wt, pintercept=1.5)
# using formula with unknowns
nordr(y, dist="prop", mu=~b0+b1*carrier, pmu=pmu, weights=wt, pintercept=1.5)
# continuation ratio
nordr(y, dist="cont", mu=mu, pmu=pmu, weights=wt, pintercept=1.5)
# adjacent categories
nordr(y, dist="adj", mu=~carrierf, pmu=pmu, weights=wt, pintercept=1.5)
#
# Haberman (1974) Biometrics 30, 589-600
# institutionalized schizophrenics: 3x3 contingency table
y <- rep(0:2,3)
fr <- c(43,6,9,
16,11,18,
3,10,16)
length <- gl(3,3)
## Not run:
# fit continuation ratio model with nordr and as a logistic model
```

```
nordr(y, mu=~length, weights=fr, pmu=c(0,-1.4,-2.3), pint=0.13,
dist="cont")

## End(Not run)
# logistic regression with reconstructed table
frcr <- cbind(c(43,16,3,49,27,13),c(6,11,10,9,18,16))
lengthord <- gl(3,1,6)
block <- gl(2,3)
summary(glm(frcr~lengthord+block,fam=binomial))
# note that AICs and deviances are different
```

---

ordglm

*Generalized Linear Ordinal Regression Models*


---

### Description

ordglm fits linear regression functions with logistic or probit link to ordinal response data by proportional odds.

### Usage

```
ordglm(
  formula,
  data = parent.frame(),
  link = "logit",
  maxiter = 10,
  weights = 1
)
```

### Arguments

formula	A model formula. The response must be integers numbered from zero to one less than the number of ordered categories.
data	An optional data frame containing the variables in the model.
link	Logit or probit link function.
maxiter	Maximum number of iterations allowed.
weights	A vector containing the frequencies for grouped data.

### Value

A list of class ordglm is returned. The printed output includes the -log likelihood, the corresponding AIC, the deviance, the maximum likelihood estimates, standard errors, and correlations.

### Author(s)

J.K. Lindsey, adapted and heavily modified from Matlab code (ordinalMLE) by J.H. Albert.

## References

- Jansen, J. (1991) Fitting regression models to ordinal data. *Biometrical Journal* 33, 807-815.  
Johnson, V.E. and Albert, J.H. (1999) *Ordinal Data Modeling*. Springer-Verlag.

## See Also

[glm](#), [nordr](#)

## Examples

```
# McCullagh (1980) JRSS B42, 109-142
# tonsil size: 2x3 contingency table
y <- c(0:2,0:2)
carrier <- gl(2,3,6)
wt <- c(19,29,24,497,560,269)
ordglm(y~carrier, weights=wt)
```

---

rs2

*Two-factor Box-Tidwell Nonlinear Response Surface Models*

---

## Description

rs2 fits a two-covariate power-transformed response surface by iterating the function, [glm](#).

## Usage

```
rs2(
  y,
  x1,
  x2,
  power = c(1, 1),
  weight = rep(1, length(x1)),
  family = gaussian,
  iterlim = 20
)
```

## Arguments

y	Response variable
x1	First covariate
x2	Second covariate
power	Initial estimates of the two power transformations
weight	Weight vector
family	glm family
iterlim	Iteration limit



**Value**

A list of class, rs, is returned containing the model and the power estimates.

**Author(s)**

J.K. Lindsey

**See Also**

[lm](#), [glm](#), [gnlr](#), [gnlr3](#), [rs3](#)

**Examples**

```
x1 <- rep(1:4,5)
x2 <- rep(1:5,rep(4,5))
y <- rpois(20,1+2*sqrt(x1)+3*log(x2)+4*x1+log(x2)^2+2*sqrt(x1)*log(x2))
rs2(y, x1, x2, family=poisson)
```

---

 rs3

*Three-factor Box-Tidwell Nonlinear Response Surface Models*

---

**Description**

rs3 fits a three-covariate power-transformed response surface by iterating the function, [glm](#).

**Usage**

```
rs3(
  y,
  x1,
  x2,
  x3,
  power = c(1, 1, 1),
  weight = rep(1, length(x1)),
  family = gaussian,
  iterlim = 20
)
```

**Arguments**

y	Response variable
x1	First covariate
x2	Second covariate
x3	Third covariate
power	Initial estimates of the three power transformations

weight	Weight vector
family	glm family
iterlim	Iteration limit

**Value**

A list of class, `rs`, is returned containing the model and the power estimates.

**Author(s)**

J.K. Lindsey

**See Also**

[lm](#), [glm](#), [gnlr](#), [gnlr3](#), [rs2](#)

**Examples**

```
x1 <- rep(1:4,5)
x2 <- rep(1:5,rep(4,5))
x3 <- c(rep(1:3,6),1,2)
#y <- rpois(20,1+2*sqrt(x1)+3*log(x2)+1/x3+4*x1+log(x2)^2+1/x3^2+
# 2*sqrt(x1)*log(x2)+sqrt(x1)/x3+log(x2)/x3)
y <- c(9,11,14,33,11,19,20,27,22,32,24,24,20,28,25,41,26,31,37,34)
rs3(y, x1, x2, x3, family=poisson)
```

# Index

## \* models

- bnlr, 2
- fit.dist, 4
- fmr, 6
- gnlr, 10
- gnlr3, 14
- nlr, 18
- nordr, 20
- ordglm, 23
- rs2, 24
- rs3, 25

bnlr, 2

finter, 3, 8, 12, 16, 17, 19, 22

fit.dist, 4

fmr, 6, 12, 17, 19, 22

glm, 3, 8, 12, 17, 19, 22, 24–26

glmm, 12, 19, 22

gnlmix, 12

gnlmm, 12, 19, 22

gnlr, 3, 8, 10, 17, 19, 22, 25, 26

gnlr3, 3, 8, 12, 14, 19, 22, 25, 26

lm, 8, 12, 17, 19, 25, 26

lvna, 2, 6, 10, 15, 18, 21

nlm, 3, 8, 12, 16, 19, 21

nlr, 12, 17, 18, 22

nls, 12, 17, 19

nordr, 20, 24

ordglm, 22, 23

restovec, 2, 6, 10, 15, 18, 21

rmna, 2, 6, 10, 15, 18, 21

rs2, 24, 26

rs3, 25, 25